

Competence Manager—Applying the Semantic Web in Practice

Mikko Laukkanen and Heikki Helin

TeliaSonera Finland

P.O. Box 970 (Teollisuuskatu 13), FIN-00051 Sonera, Finland

{mikko.laukkanen,heikki.j.helin}@teliasonera.com

Abstract

A corporate memory is an explicit, disembodied, and persistent representation of knowledge and information in an organization. Competence management is an essential part of a corporate memory. In this paper we introduce a Competence Manager, which allows managing and publishing competencies and other relevant resources about a set of topics. The knowledge base in the Competence Manager is implemented using ontologies. Thus, in this paper we demonstrate, how the Semantic Web technologies can be used in a practical application on the area of competence management. The front-end of the Competence Manager is accessed using a Web-based interface, which connects to the back-end system through a Web service interface. The Competence Manager provides operations for querying the knowledge base, managing the ontologies, and providing reports about the resources stored the knowledge base.

1 Introduction

A corporate memory is an explicit, disembodied, and persistent representation of knowledge and information in an organization [Dieng *et al.*, 1999]. The corporate memory allows access to, and reuse of the organizations's knowledge, which is a sum of the knowledge of its members. The corporate memory can be best characterized as being distributed and heterogeneous [Gandon *et al.*, 2002]: the information landscape of the corporate memory, the population of users (employees) of the memory within the organization, and tasks to be performed on corporate memory are all by nature distributed and heterogeneous.

Competencies cover only a part of a corporate memory, but in our opinion the competence management is a good starting point and an essential building block in a comprehensive corporate memory management system. In this paper we describe an ontology-based corporate memory tool called Competence Manager, with which the members of an organization (e.g., a company) are able to manage and publish their competencies and other relevant resources about a set of topics. Once published, other users are able

to make queries to the Competence Manager to find out suitable persons and/or other resources for instance when forming a project. The difference to other typical competence management tools is that in the case where the best candidates (i.e., persons having competence directly on the searched topic) are not available, the candidates having competence on nearby topic(s) can be found. Also, by using ontologies in the Competence Manager allows supporting multiple domain ontologies, which can be imported at runtime. In addition to maintaining multiple simultaneous domain ontologies, the portability—for instance from one organization to another—does not require changes to the implementation.

In [Berners-Lee *et al.*, 2001], the authors define that “the Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation”. The Competence Manager aims at realizing this definition in the area of corporate memory and especially the competence management part of it. The knowledge base in the Competence Manager is modeled as ontologies; thus, in this paper we demonstrate, how the Semantic Web technologies can be used in a practical application on the area of competence management.

The rest of this paper is organized as follows. Section 2 gives the motivation for our work, and describes the tasks the Competence Manager can be used for. Section 3 introduces the general architecture of the Competence Manager. Sections 4 and 5 describe the Competence Manager functionality; we explain what kind of ontologies the Competence Manager uses, and how the Competence Manager is used in practice. Finally, Section 6 concludes this paper.

2 Motivation

For companies with intellectual property it is crucial to have an environment, where the knowledge can be shared efficiently within the company. One way of sharing knowledge—and maybe the most common one—is to first establish a network of contacts; that is, an employee knows, what kind of competencies her co-workers have. After that, the knowledge is shared by asking the competent person directly.

As companies get larger, it becomes increasingly difficult to manage the knowledge and competencies that their

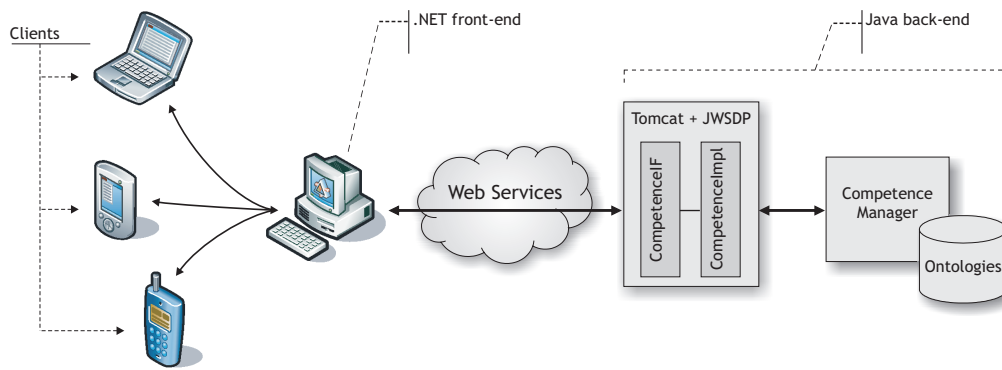


Figure 1: Competence Manager architecture

employees have. The knowledge sharing within small companies usually happens in a face-to-face fashion between the employees; everyone knows each other, and the competencies of their co-workers. However, in large companies, which are usually geographically distributed even between different countries, the contact network of an employee usually covers only a fraction of the whole company. In these kind of companies the efficient knowledge sharing is extremely challenging.

The Competence Manager is targeted at managing and sharing competencies, and it can be used at least for the following crucial tasks. Firstly, the Competence Manager can be used to track down competent persons within a company. This functionality could be used when a person needs assistance or help for something, a project manager is gathering competent persons for a project, or a document writer needs some background information about something. Secondly, the Competence Manager can be used as a tool during development discussions between superiors and their employees; the current competencies and the competence areas needing improvements can be identified, and a development path(s) can be planned. Thirdly, the Competence Manager can serve as a tool for company management in strategic coordination activities. The overall competence levels on various areas can be identified, and in the case of defining company's strategic goal, the Competence Manager can be used in identifying the areas needing more or less competence.

The users access the Competence Manager through a front-end server, which takes care rendering the user interface. The front-end server uses Web service interface in order to access the ontology server, which maintains the competence ontologies and implements the inference engine operating on those ontologies. In the following subsections we will describe these in more detail.

3 Architecture

The architecture and the general parts of the Competence Manager are depicted in Figure 1. The users (clients) access the system through a front-end server, which renders the user interface according to the client device. The front-end is implemented as Microsoft .NET Web application, because it allows an easy and fast way to develop graphical user interfaces that are able to adapt to a variety of client devices.

The back-end of the Competence Manager is exposed as a Web service, which the front-end server accesses in delivering the requests from the user interface. One reason for choosing the .NET for implementing the front-end is its built-in support for Web services. However, the front-end could be implemented with other technologies as well, such as Java-based application servers. Furthermore, due to the Web service interface, the Competence Manager could be accessed by Web service-enabled clients or devices directly.

The Java Web Service Development Pack (JWSDP) 1.4 is used to implement the Web service interface and the implementation, and the Tomcat 5.0 application server hosts the deployed Web service. The Web service implementation forwards all the requests to the Competence Manager, which implements the real functionality. This loose coupling between the Web service implementation and the Competence Manager makes it easy to switch the JWSDP to another Web service application server, if desired. One reason for choosing the JWSDP over for instance Apache Axis is that there is a client-side Web service API targeted at J2ME-capable mobile phones, and this API is built on top of JWSDP. This enables accessing the Competence Manager directly from a Midlet running on a mobile phone.

The Competence Manager maintains the ontologies and implements the inference engine used for instance in queries. We will describe these in more detail in the next sections.

4 Ontologies in the Competence Manager

The ontologies in Competence Manager are divided into general and domain-specific parts. The general ontology defines the mandatory classes and properties for the competence topic hierarchy. The domain ontologies define the domain-specific resources, and must conform to the general ontology. This asserts that the inference engine works correctly with the domain ontologies.

Examples of the classes defined in the general ontology include `Topic`, `Resource`, `Group`, and `Relevance`. These classes have properties such as `hasParentTopic`, `hasResource`, `hasMember`, `relevanceTopic`, and `relevanceValue`. These are depicted in Figure 2. The topic hierarchy in the domain-specific ontology is realized as instances of the `Topic` class, and parent-child relations are defined using the `hasParentTopic` property. The

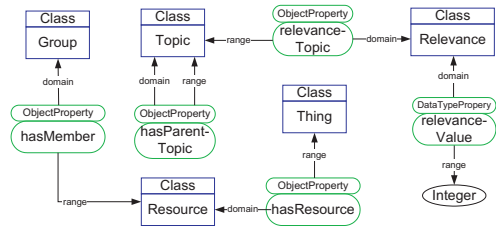


Figure 2: General ontology

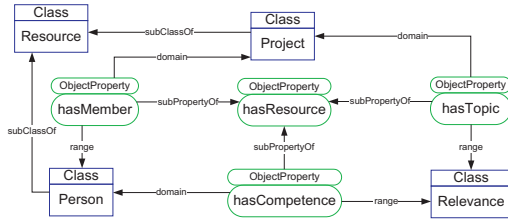


Figure 3: Sample domain ontology

instances of the *Relevance* class specify the relevance level of some resource (e.g., a person’s competence) to a certain topic. The relevance value in the Competence Manager can get integer values from one to five. For instance, an employee having an excellent knowledge on some topic may set the relevance value to five, whereas some other employee with only some background information about the same topic could set the value to one. The *Group* class allows tying a group of resources together. This is useful for instance when modeling static organizational units within a company, but also when creating cross-organizational groups of competent persons dynamically.

The domain ontology defines the classes and their instances that have competencies or are related to the topics. The resources can be freely defined as domain-specific classes, as long as they are subclasses of the *Resource* class. Furthermore, the relationships (properties) between the domain-specific resources can be freely defined, but the properties must be defined as a sub-property of the *hasResource* property. For instance, the domain-specific ontology depicted in Figure 3 depicts that a *Person* has competence of a *Topic* (with some *Relevance* level), and that the *Person* can be a part of a project, which is related to some *Topic*.

Figure 4 depicts a sample fragment of instances for the domain ontology about software technologies. In this case a rectangle represents an instance, and the identifier inside “<” and “>” characters denote the class, which the instance realizes. The properties follow the same scheme as in Figure 2. The topics are on the lefthand side, while the resources are on the right. The instance “Mikko” of a *Person* class has competencies on topics “OWL” and “Java”. The instances of *Project* and *Document* classes are about some topic, and also may have properties binding them together, such as *hasDocument*, *memberOf*, and *authorOf*.

5 Using the Competence Manager

The operations the Competence Manager supports can be divided into two parts. The browsing interface can be

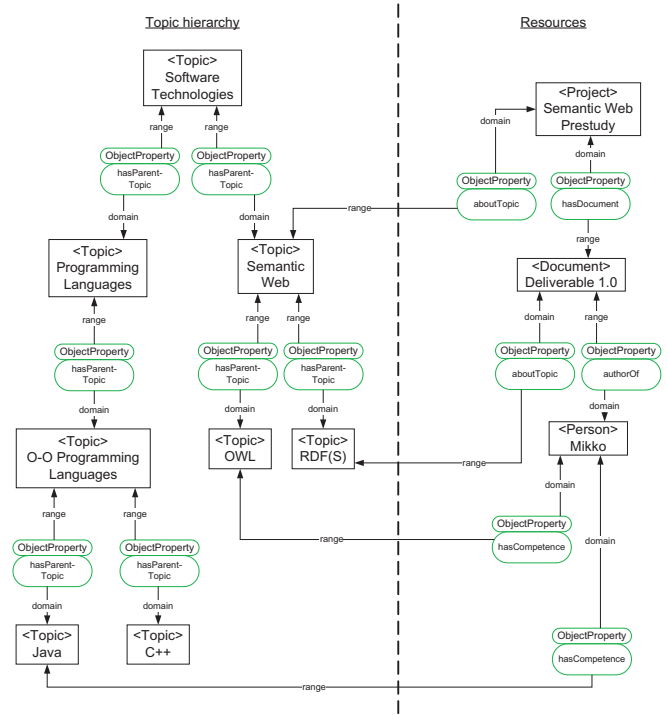


Figure 4: Sample instances of topics and other resources

used to make queries to the Competence Manager and display (graphical) reports about the information stored in the knowledge base. The management interface on the other hand can be used to manage the domain ontologies, and it supports functionality to import new ontologies and modifying the existing ones. In the following subsections we will cover these operations shortly.

5.1 Managing Domain Ontologies

The management interface allows importing new domain ontologies to the Competence Manager, and managing the existing ones. The Competence Manager supports multiple domain ontologies, which can be imported at runtime. In addition to maintaining multiple simultaneous domain ontologies, the portability—for instance from one organization to another—does not require changes to the implementation. The only thing that has to be done is to define the domain ontology, and import it to the Competence Manager.

The management of the existing ontologies is about adding, modifying, and deleting the topics and resources of the domain ontology. For instance, after attending a course on Java programming language, the employee may increase her competence on Java to the upper level, and optionally add to the competence description about the things learned on the course. Respectively, other kind of domain-specific resources—such as persons, documents, and projects—can be managed for instance by linking people and documents to projects.

5.2 Making Queries

The Competence Manager provides three ways of making queries to the knowledge base. The regular expression-enabled searches (see Figure 5, search area) can be used

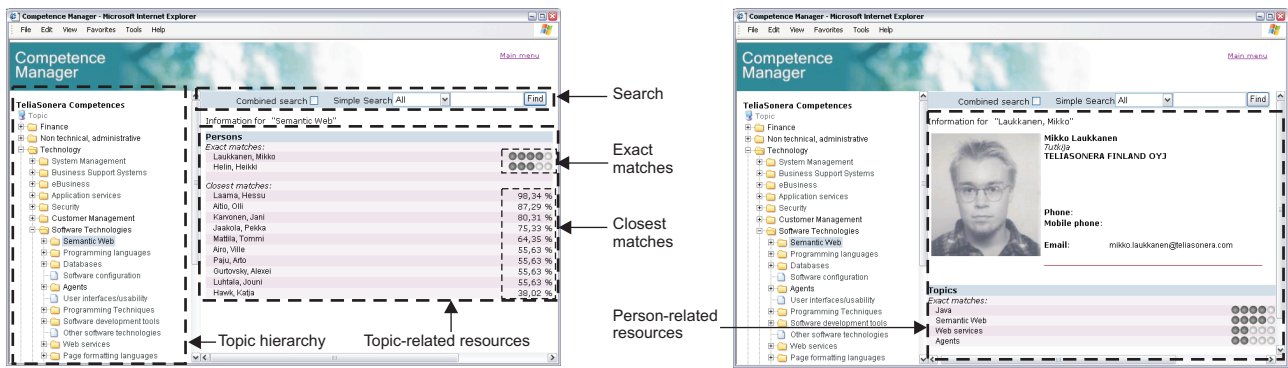


Figure 5: Using the Competence Manager user interface

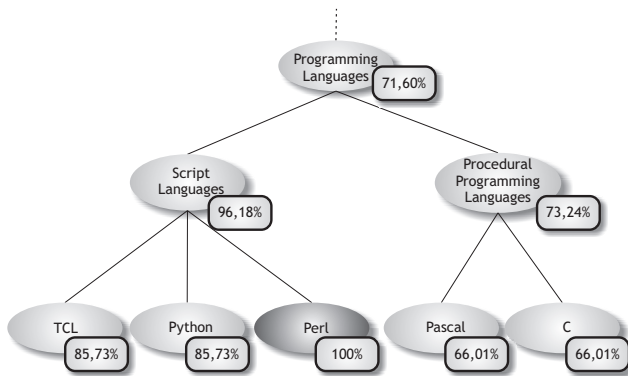


Figure 6: Closest matches when searching for topic “Perl”

to find some specific resources. These kind of searches do not make use of any ontological relationships, but apply the search on all the instances in the knowledge base. Once the resource(s) has been found, the topics and other related resources can be retrieved (see Figure 5, person-related resources areas).

The second kind of query takes a topic as a parameter, and returns all the related resources found in the knowledge base. In this case only the exact matches—that is, resources referring to that specific topic in the query—are displayed. For the exact matches, the knowledge level is displayed as bullets ranging from one to five (see Figure 5, exact matches area). If there are no exact matches, the third kind of query allows querying the resources that are closest to the queried topic (see Figure 5, closest matches area). This query makes use of an algorithm called object match [Stojanovic *et al.*, 2001], which calculates the distance to the neighboring topics, and returns the resources related to the closest ones. This is useful in a case where the best candidates (i.e., the persons having competence directly on the searched topic) are not available, but some candidates having competence on nearby topic(s) are found. For instance, when competent persons or other resources on the searched topic “Perl” are not available, the persons having competence on—or the resources related to—topics “TCL”, “Python”, or even “C”, which does not have the same direct parent topic with the others, might be close enough to the searched topic, and they could be still accepted. This is depicted in Figure 6, where the closest topics (with percentages) to “Perl” are shown. The algorithm

takes into account the depth of the queried topic in the topic hierarchy. For instance, the topics “C” and “Pascal” (located at some leaf nodes), are semantically closer to each other than top-level topics “Marketing” and “Technology”.

6 Conclusion

In this paper we demonstrated, how the Semantic Web technologies can be used in a practical application. We described an ontology-based competence management tool called Competence Manager, which can be used in searching for competent persons within a company, as a tool during development discussions between superiors and their employees, and as a tool for company management in strategic coordination activities. The Competence Manager provides operations for querying the knowledge base, managing the ontologies, and providing reports about the resources stored in the knowledge base. The Competence Manager is not currently publicly available.

References

- [Berners-Lee *et al.*, 2001] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, May 2001.
- [Dieng *et al.*, 1999] Rose Dieng, Olivier Corby, Alain Giboin, and Myriam Ribière. Methods and Tools for Corporate Knowledge Management. *International Journal of Human-Computer Studies, Special Issue on Knowledge Management*, 51:567–598, 1999.
- [Gandon *et al.*, 2002] Fabien Gandon, Laurent Berthelot, and Rose Dieng-Kuntz. A Multi-Agent Platform for a Corporate Semantic Web. In Maria Gini, Toru Ishida, Cristiano Castelfranchi, and W. Lewis Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS’02)*, pages 1025–1032. ACM Press, July 2002.
- [Stojanovic *et al.*, 2001] Nenad Stojanovic, Alexander Maedche, Steffen Staab, Rudi Studer, and York Sure. SEAL: A Framework for Developing SEMantic PortALS. In *Proceedings of the International Conference on Knowledge Capture*, pages 155–162, Victoria, British Columbia, Canada, October 2001. ACM Press.